work queue and each of the end-to-end contexts portions at the source endnode to the receive work queue and the corresponding end-to-end context portion at each of the destination endnodes.

3. The distributed computer system of claim 2 wherein the source endnode includes a network interface controller which packetizes the message data into frames.

4. The distributed computer system of claim 3 wherein the destination endnodes each include a network interface controller which acknowledges receipt of frames multicast from the source endnode.

5. The distributed computer system of claim 4 wherein the network interface controller and the end-to-end context portion in each destination endnode ensure strong ordering of received frames multicast from the source endnode, such that the frames are received in a same defined order as transmitted from the source endnode.

6. The distributed computer system of claim 4 wherein the source endnode retransmits frames that are not successively acknowledged in the reliable multicast.

7. The distributed computer system of claim 3 wherein the network interface controller in the source endnode includes hardware which replicates frames to be provided in the series of unicasts.

8. The distributed computer system of claim 2 wherein the source endnode includes software verbs which perform the series of unicasts as a series of individual sequenced message send operations.

9. The distributed computer system of claim 2 wherein changes in composition of the endnodes participating in the multicast group are communicated to all endnodes participating in the multicast group.

10. The distributed computer system of claim 2 wherein the source endnode and each destination endnode maintains a list of destination addresses for all other endnodes participating in the multicast group.

11. The distributed computer system of claim 4 wherein the network interface controller in each destination endnode generates cumulative acknowledgments.

12. The distributed computer system of claim 4 wherein the network interface controller in each destination endnode generates acknowledgments on a per frame basis.

13. The distributed computer system of claim 4 the network interface controller of the source endnode includes a completion processing unit which gathers acknowledgements from the destination endnodes and completes frame operation by informing the source process of an operation status of multicast frames.

14. The distributed computer system of claim 13 wherein the source endnode further comprises:

a completion queue containing information related to completed work queue elements, wherein the completion processing unit communicates with the source process via the completion queue.

15. The distributed computer system of claim 13 wherein the completion processing unit informs the source process which destination processes, if any, did not receive multicast frames.

16. The distributed computer system of claim 13 wherein the completion processing unit includes an acknowledgement counter which counts acknowledgements received from the corresponding destination endnodes in the multicast group indicating that the corresponding destination endnode has received a frame multicast from the source endnode.

17. The distributed computer system of claim 16 wherein completion processing unit generates a completion event to the source process when the acknowledgement counter

indicates that a predetermined percentage of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

18. The distributed computer system of claim 16 wherein completion processing unit generates a completion event to the source process when the acknowledgement counter indicates that all of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

19. The distributed computer system of claim 13 wherein the completion processing unit includes a bit-mask array which assigns a unique bit for each destination endnode in the multicast group and clears each bit as a corresponding acknowledgment is received from the corresponding destination endnode in the multicast group indicating that the corresponding destination endnode has received a frame multicast from the source endnode.

20. The distributed computer system of claim 19 wherein the completion processing unit generates a completion event to the source process when the bit-mask array has a predetermined percentage of bits cleared in the bit-mask array indicating that that a predetermined percentage of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

21. The distributed computer system of claim 19 wherein the completion processing unit generates a completion event to the source process when the bit-mask array has all bits cleared in the bit-mask array indicating that that all of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

22 The distributed computer system of claim 13 wherein the completion processing unit includes a timing window, wherein expiring of the timing window without necessary conditions for a completion event for a corresponding multicast frame occurring indicates that any missing acknowledgments are to be tracked and resolved.

23. The distributed computer system of claim 2 wherein a given proc ss joins the multicast group by performing a multicast join operation.

24. The distributed computer system of claim 2 wherein a given process leaves the multicast group by performing a multicast leave operation.

25. A method of multicasting in a distributed computer system including a source endnode participating in a multicast group and multiple destination endnodes participating in the multicast group, the method comprising:

producing message data with a source process at the source endnode;

describing the message data for multicasting with work queue elements in a send work queue at the source endnode;

describing where to place incoming message data with work queue elements in a receive work queue at each of the multiple destination endnodes;

storing in each of multiple end-to-end contexts state information at the source node and state information at a corresponding one of the destination endnodes to ensure the reception and sequencing of message data multicast from the source endnode to the corresponding one of the destination endnodes; and

reliably multicasting data including performing a series of replicated unicasts of message data though the send work queue and each of portions of the end-to-end contexts at the source endnode to the receive work queue and corresponding end-to-end context portions at each of the destination endnodes.

26. The method of claim 25 further comprising:

packetizing, at the source endnode, the message data into frames.

27. The method of claim 26 further comprising:

acknowledging, at each of the destination endnodes, receipt of frames multicast from the source endnode.

28. The method of claim 27 further comprising:

ensuring strong ordering of received frames multicast from the source endnode, such that the frames are received in a same defined order as transmitted from the source endnode.

29. The method of claim 27 further comprising:

retransmitting frames that are not successively acknowledged in the reliable multicast.

30. The method of claim 26 wherein the packetizing the message data into frames includes replicating replicatings frames to be provided in the series of unicasts.

31. The method of claim 25 wherein the series of unicasts are performed as a series of individual sequenced message send operations.

32. The method of claim 25 further comprising:

communicating changes in composition of the endnodes participating in the multicast group to all endnodes participating in the multicast group.

33. The method of claim 25 further comprsing:

maintaining, at the source endnode and each destination endnode, a list of destination addresses for all other endnodes participating in the multicast group.

34. The method of claim 27 wherein the acknowledging, at each of the destination endnodes, includes generating cumulative acknowledgments.

35. The method of claim 27 wherein the acknowledging, at each of the destination endnodes, includes generating acknowledgments on a per frame basis.

36. The method of claim 28 further comprising:

gathering, at the source endnode, acknowledgements from the destination endnodes; and

completing frame operation by informing the source process of an operation status of multicast frames.

37. The method of claim 36 further comprising:

maintaining information related to completed work queue elements in a completion queue; and

communicating with the source process via the completion queue.

38. The method of claim 36 further comrprising:

informing the source process which destination processes, if any, did not receive multicast frames.

39. The method of claim 36 further comprising:

counting acknowledgements received from the corresponding destination endnodes in the multicast group indicating that the corresponding destination endnode has received a frame multicast from the source endnode.

40. The method of claim 39 further comprising:

generating a completion event to the source process when the counted acknowledgements indicate that a predetermined percentage of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

41. The method of claim 39 further comprising:

generating a completion event to the source process when the counted acknowledgements indicate that all of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

42. The method of claim 36 further comprising:

assigning a unique bit in a bit-mask array for each destination endnode in the multicast group; and

clearing each bit in the bit-mask array as a corresponding acknowledgment is received from the corresponding destination endnode in the multicast group indicating that the corresponding destination endnode has received a frame multicast from the source endnode.

43. The method of claim 42 further comprising:

generating a completion event to the source process when a predetermined percentage of bits are cleared in the bit-mask array indicating that that a predetermined percentage of the

destination endnodes in the multicast group have acknowledged the multicast frame has been received.

44. The method of claim 42 further comprising:

generating a completion event to the source process when all bits are cleared in the bit-mask array indicating that that all of the destination endnodes in the multicast group have acknowledged the multicast frame has been received.

45. The method of claim 36 further comprising:

maintaining a timing window at the source endnode, wherein expiring of the timing window without necessary conditions for a completion event for a corresponding multicast frame occurring indicates that any missing acknowledgments are to be tracked and resolved.

46. The method of claim 25 further comprising:

performing a multicast join operation to join a given process to the multicast group.

47. The method of claim 25 further comprising:

performing a multicast leave operation to remove a given process from the multicast group.